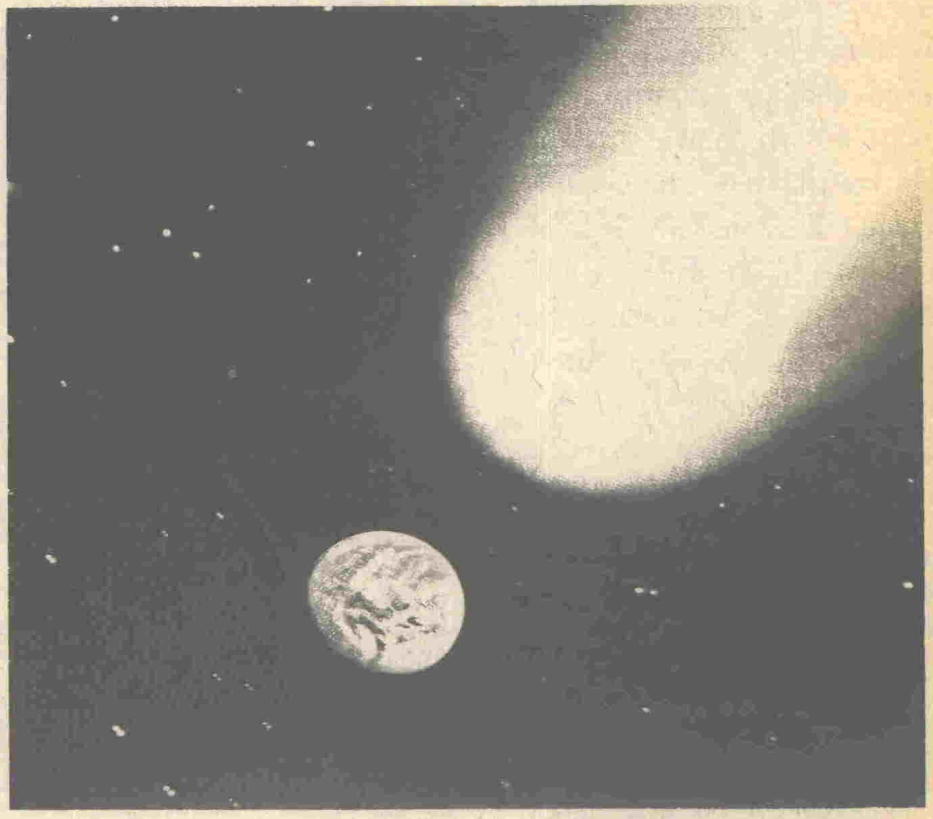
การเขียนโปรแกรม Ball moving game

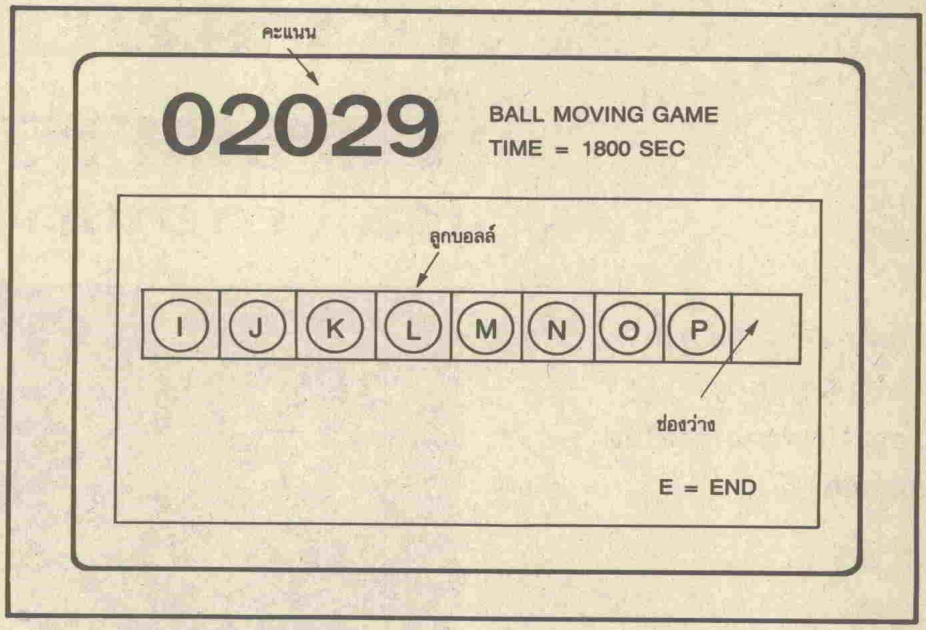
เขียนเกมด้วยปาสคาล โดยอาศัยหลักการเขียนโปรแกรมบนเครื่องเวอร์กสเตชัน



ไม่ได้เขียนบทความที่เกี่ยวกับเกมคอมพิวเตอร์มาลงวารสารนี้เสียหลายเดือน จนทำให้ลูกค้าบางคนโทรไปถามที่บ้านว่าหมดแรงคิดแล้วหรืออย่างไร คำตอบคือ "ไม่" แต่หมู่นี้ไปยุ่งกับงานประเภทอื่น ๆ อยู่มาก ก็เลยไม่ค่อยมีเวลาจะมานั่งที่หน้าจอ และในความจริงแล้วหมู่นี้คิดเกมได้เยอะแยะเพียงแต่ว่า "ไม่ค่อยมีเวลาจะทำ" เมื่อวานนี้พอดีมีเวลาว่างอยู่หน่อยหนึ่ง ก็เลยแวะไปคุยกับลูกทีมที่เขาศึกษาและเล่นกับเครื่อง SPARC ของบริษัทซัน-ไมโครซีสเต็ม ผมได้ให้การบ้านเขาไปคิดเล่น ๆ คือให้ลองทำเกม "Ball Moving" ขึ้นมาให้ดู ผมอยากจะดูว่าการพัฒนาโปรแกรมลงบนเครื่องประเภทไฮเพาเวอร์เวอร์กสเตชัน ด้วยภาษาซีนั้นจะเร็วแค่ไหน ผมเชื่อว่าเขาน่าจะทำได้ใน 2-3 วัน แต่ดูพวกเขาจะไม่ค่อยเชื่อมั่นตัวเองเท่าใดนัก และเพื่อเป็นการยืนยันว่างานแบบนี้มันง่ายนิดเดียว ผมกลับไปบ้านแล้วก็เลยลองพัฒนาด้วยปาสคาล (เวอร์ชัน 3.0) ที่ผมถนัด แล้ววันนี้ก็เลยเอามาสาธิตให้ดู ผมคิดว่าท่านผู้อ่านก็คงจะสนใจว่าเกมนี้มันเป็นอย่างไรและทำไมมันจึงง่ายต่อการพัฒนา

เกม "Ball Moving" นี้มีหลักการง่าย ๆ คือ สมมติว่าเรามีช่องบรรจุลูกบอลอยู่ 9 ช่อง ช่องทั้ง 9 จะวางเรียงกันจากซ้ายไปขวา และเรามีลูกบอลอยู่ 8 ลูก ลูกบอลเหล่านี้จะมีตัวอักษรติดอยู่ และเป็นตัวอักษรภาษาอังกฤษที่เรียงกัน ลูกบอลทั้ง 8 จะบรรจุอยู่ในช่องที่ 1 ถึง 8 ตามลำดับ นี่เป็นสิ่งที่เราจะต้องวาดขึ้นมาบนจอภาพเสียก่อน ต่อจากนั้นมาเราก็จะให้เครื่องมันจัดการย้ายลูกบอลเหล่านี้ไปมา การย้ายนี้ต้องทำทีละลูกและมันจะต้องยกไปลงในช่องที่ว่างอยู่ ท่านผู้อ่านคงพอจะนึกภาพออก คือมันจะมีช่องว่างอยู่เพียงช่องเดียว เมื่อย้ายไปแล้วก็จะเกิดช่องว่างขึ้น ณ จุดที่เราเพิ่งจะย้ายออกไป การย้ายโดยเครื่องคอมพิวเตอร์นี้ให้ทำอย่างแรนด้อม (Randomly) และทำเป็นจำนวน 100 ครั้ง เมื่อทำเสร็จก็จะพบว่าตัวอักษรบนลูกบอลเหล่านั้นจะสลับที่กันหมด หน้าที่ของผู้เล่นเกมก็คือ ต้องย้ายลูกบอลให้กลับไปอยู่เรียงกันจากช่อง 1 ถึง 8 เหมือนเดิม ผู้เล่นจะต้องพยายามใช้

จำนวนครั้งของการย้ายลูกบอลให้น้อยที่สุด ทั้งนี้เพราะการย้ายครั้งแรกจะหักคะแนน 15 คะแนน และการย้ายครั้งต่อ ๆ ไปจะหัก 1.5 เท่าของครั้งที่ผ่านมา ในการที่จะบอกเครื่องว่าจะย้ายลูกบอลลูกไหนนั้นให้กดแป้นตัวอักษรที่ตรงกับอักษรบนลูกบอลนั้น คะแนนเริ่มต้นให้มีค่าเท่ากับ 2029 (เท่ากับปีที่คนเหล็กย้อนอดีตมา) ตัวอักษรบนลูกบอลนั้นจะเริ่มจากตัว F หรือ G หรือ H ก็ได้ เราจะต้องให้เครื่องมันเลือกอย่างแรนด้อม เมื่อเลือกแล้วก็จะใช้ 8 ตัวอักษรติดต่อกัน เช่น ถ้าเริ่มที่ I ก็ต้องเป็น I, J, K, L, M, N, O, P ตามลำดับ ที่ผมให้เลือกตั้งแต่ F ก็เพราะตัว E จะเอาไว้ใช้บอกให้จบเกม (E = End) รูปที่ 1 แสดงต้นร่างของหน้าจอภาพเมื่อยังไม่มีการโยกย้ายลูกบอล



รูปที่ 1 ภาพเริ่มต้นของเกม Ball Moving

เอ้อ เกือบลืมจุดสำคัญไป เกมนี้ให้เวลาทั้งหมด 1800 วินาที (ครึ่งชั่วโมง) เมื่อเราจัดเรียงกลับเข้าที่ได้แต่ละครั้งก็จะให้คะแนน 1250 แต้ม สิ่งที่ท่านต้องระวังเป็นพิเศษคือ เล่นรอบแรก ๆ นั้นผู้เล่นจะต้องพยายามอย่าให้ขาดทุน ทั้งนี้เพราะถ้าคะแนนติดลบจะจบเกมไปเลย นอกจากนี้แล้วก็ต้องระวังการกดแป้นให้ดี กดแป้นผิดจะหักคะแนน 30 คะแนนต่อ 1 ครั้ง

ครับ จากโจทย์ที่กล่าวมาข้างบนนี้เราก็ต้องมาวางลอจิกของโปรแกรม ผมจะขอพูดถึงรูทีนที่สำคัญ ๆ สัก 3-4 รูทีน แล้วก็พูดถึงเมนลูปเสียหน่อยหนึ่ง ท่านผู้อ่านคงต้องดูรายละเอียดอื่น ๆ เอาจากลิสติง สำหรับตัวโปรแกรมใช้บริการทางซอฟต์เมล์ได้เลยครับ

หลักการที่สำคัญข้อที่ 1 ก็คือเราจะสร้างรูปภาพช่องสี่เหลี่ยมจำนวน 9 ช่อง และมีลูกบอลอยู่ภายในได้อย่างไร? อันนี้ท่านต้องดูที่โพรซีเดอร์ที่ชื่อว่า GENCHAR; ผมเริ่มต้นด้วยคำสั่งว่า SHFT:= 5 + RANDOM(14); เจ้าตัวแปรที่ชื่อว่า SHFT นี้คือตัวบอกว่าเราจะเอาตัวอักษรตัวไหนเป็นตัวเริ่มต้นอักษร A, B, C,... นั้นจะตรงกับรหัสแอสกีที่มีค่าเท่ากับ 65, 66, 67,... ตามลำดับ ดังนั้น 64 + K1 จะตรงกับ A, B, C,... เมื่อ K1 มีค่าเป็น 1, 2, 3,... แต่โจทย์บอกว่าเราอยากจะให้ไปเริ่มต้นหลังตัว E ผมก็เลยเอา 5 + RANDOM(14) มาบวกเข้าไป เจ้าค่า RANDOM(14) นั้นจะมีค่าที่เป็นไปได้คือ 0, 1, 2, 3,...13 ในกรณีที่ค่านี้ไปตกที่เลข 13 ก็จะทำให้ 64 + K1 + SHFT ไปตกที่ตัว Z พอดี (อักษรตัวสุดท้ายคือ 64 + 8 + 5 + 13 = 64 + 26 = Z)

หลังจากวางแผนเรื่องการกำหนดตัวอักษรบนลูกบอลแล้วผมก็มาดูว่าผมจะเขียนช่องสี่เหลี่ยมต่าง ๆ ได้อย่างไร เรื่องนี้ผมสร้างโพรซีเดอร์เอาไว้ในชื่อว่า DRAWBOX(X1, Y1, X2, Y2, C: INTEGER); เจ้า (X1, Y1) คือจุดมุมบนด้านซ้าย ส่วน (X2, Y2) จะเป็นจุดมุมล่างด้านขวา เมื่อกำหนดจุด 2 มุมนี้ให้แล้วเราก็สามารถลากเส้น 4 เส้นเป็นด้านทั้ง 4 ได้ ส่วนตัว C นั้นคือรหัสสีของเส้น เจ้าช่องสี่เหลี่ยมดังกล่าวนี้จะวางต่อเนื่องกันไป ระยะห่างจะเป็น 24 จุด (Pixel) ผมจึงบอกว่า X1:=24 * (K1−1) + 47; Y1:=89; และ X2:=X1 + 24; Y2 = Y1 + 24; ด้วยวิธีนี้ผมสามารถเขียนช่องสี่เหลี่ยมต่อเนื่องกัน 9 ช่อง โดยเปลี่ยนค่า K1 จาก 1 ถึง 9 สำหรับตัวลูกบอลนั้นผมใช้คำสั่งว่า CIRCLE(X1 + 12, Y1 + 12, 12, 1); ทั้งนี้เพราะจุดศูนย์กลางของแต่ละช่องจะอยู่ที่ (X1 + 12, Y1 + 12) ผมใช้รัศมี = 12 ทั้งนี้เพื่อให้ลูกบอลวาดลงฟิตในช่องสี่เหลี่ยมนี้พอดี นอกจากจะเขียนภาพลูกบอลแล้ว

ผมก็ใช้คำสั่ง FILLSHAPE (X1 + 13, Y1 + 13, 3, 1); เพื่อระบายสีของลูกบอล เมื่อระบายสีแล้วก็สั่ง WRITE (CH); ลงไปบนลูกบอลนั้น ตัวอักษร CH จะตรงกับรหัสแอสกีที่กล่าวถึงในตอนต้น เมื่อเขียนตัวอักษรบนลูกบอลแล้วก็ต้องจดเอาไว้ด้วยว่าช่องนี้มีอะไรบรรจุอยู่ เจ้าตัวเก็บรหัสนี้คือ TAB [K1] มันเป็นอะเรย์ขนาด 9 ช่อง แต่ท่านควรสังเกตว่า เมื่อ K1 = 9 นั้นเราต้องการเว้นให้เป็นช่องว่าง ผมจึงใช้คำสั่งว่า IF K1 = 9 THEN BEGIN GH:=' '; TAB [K1]:=0; END; หลังจากเขียนตัวอักษรลงบนลูกบอลแล้วเราต้องใช้คำสั่ง DRAWBOX (X1 + 8, Y1 + 6, X1 + 16, Y1 + 14, 0); เข้าไปด้วยอันนี้เพื่อทำกรอบสีดำรอบตัวอักษร ถ้าไม่ตัดขอบตัวอักษรจะทำให้อ่านยากขึ้น

โอเค ด้วยโพรซีเดอร์ GENCHAR; และ DRAW BOX(...); ผมสามารถวาดรูปเป็นช่องสี่เหลี่ยมติดต่อเนื่องกันและภายในช่องจะมีลูกบอลอยู่ช่องละ 1 ลูก ยกเว้นช่องที่ 9 ที่นี้ก็จะมาถึงเรื่องของการย้ายลูกบอล อันนี้ทำได้โดยโพรซีเดอร์ที่ชื่อว่า MOVEBALL (L1, L2: INTEGER); เจ้า L1 กับ L2 คือตัวบอกว่าจะย้ายจากช่องไหนไปช่องไหน L1 คือตัวลูกบอล L2 คือช่องว่างที่จะย้ายไปหา การย้ายลูกบอลนี้ทำได้โดยคำสั่ง GETPIC (ST1, X1, Y1, X2, Y2); และ PUTPIC (ST1, X6, Y6); ตัวแปร ST1 เป็นเนื้อที่สำหรับเก็บรูปภาพค่า (X1, Y1) และ (X2, Y2) คือจุดมุมบนซ้ายกับมุมล่างขวาของบล็อกสี่เหลี่ยมที่เราต้องการเก็บภาพขึ้นไป ส่วนค่า (X6, Y6) คือจุดมุมล่างด้านซ้ายของบล็อกสี่เหลี่ยมที่เราจะย้ายไปลง หลังจากย้ายลูกบอลไปแล้วเราก็ต้องเคลียร์ช่องเก่าให้เป็นช่องว่าง อันนี้ทำโดยคำสั่ง PUTPIC (STO, X3, Y3-1); เจ้า STO นี้คือเนื้อที่ที่เราเก็บภาพช่องว่างเอาไว้ ภาพ STO นี้มาจากไฟล์ "SHBD01" ซึ่งเป็นที่เก็บภาพตัวเลขที่ใช้แสดงคะแนน (SCORE) เราโหลดภาพเหล่านี้ในโพรซีเดอร์ชื่อ LOADPICTURE; หลังจากย้ายภาพแล้วเราต้องปรับค่าของ TAB[.] ด้วย คือ TAB[L2] จะมาจาก TAB[L1] เสร็จแล้วต้องเคลียร์ TAB[L1] ให้เป็น 0 ทั้งนี้เพื่อบอกว่าตอนนี้ตำแหน่งที่ L1 นี้เป็นช่องว่างแล้ว สำหรับโพรซีเดอร์ที่ใช้สลับที่ลูกบอลไปมาอย่างแรนด้อม คือ SCRAMBLECHAR; ในนั้นผมใช้คำสั่งว่า

```
REPEAT
    R1: = RANDOM(9) + 1;
    UNTIL TAB[R1] > 0;
```

ทั้งนี้เพื่อหาช่องที่มีลูกบอลบรรจุอยู่ และหาอย่างแรนด้อมด้วยวิธีนี้จะทำให้ย้ายลูกบอลอย่างคาดเดาไม่ออก ผลที่ได้จะทำให้ลูกบอลสลับที่กันไปหมด

มีโพรซีเดอร์ที่สำคัญอีก 2 อันที่ผมควรอธิบาย คือ โพรซีเดอร์ GETL1L2 (CH:CHAR; VAR L1, L2 : INTEGER); และ CHECKENDPATTERN; ตัวแรกนั้นมีไว้เพื่อหาค่า L1 และ L2 หรือคือ เมื่อเรากดตัวอักษรเข้าไปมันจะไปอยู่ในตัวแปร CH เราต้องหาว่าตัวนี้จะตรงกับ TAB [.] ตัวที่เท่าใด นั่นคือค่า L1 ส่วน L2 นั้นหามาได้จากการหาว่า TAB[.] ตัวใดมีค่าเท่ากับศูนย์ (0) เมื่อเราได้ค่า L1, L2 แล้วก็จะได้นำไปใช้ในโพรซีเดอร์ MOVEBALL (L1, L2 : INTEGER); สำหรับโพรซีเดอร์ CHECKENDPATTERN; นั้นมีไว้เพื่อเช็คว่าเราจัดเรียงลูกบอลกลับเข้าที่แล้วหรือยัง

ครับ หลังจากที่เขียนโพรซีเดอร์ต่าง ๆ เรียบร้อยแล้วผมก็เอามาจัดลงในเมนโปรแกรม เมนโปรแกรมเริ่มที่คำสั่งว่า SPEEDCHECK; GRAPHCOLORMODE; RANDOMIZE;... นี่เป็นการเช็คความเร็วของซีพียู ทั้งนี้เพื่อเอาไปใช้ในการทำคำสั่งพวกออกเสียงและการหน่วงให้ภาพวิ่งได้สปีดพอดี ๆ เสร็จแล้วก็เปลี่ยนจอภาพเป็นกราฟิกแบบ CGA และให้เครื่องมันเริ่มหาจุดเริ่มต้นของตัวสร้าง "RANDOM NUMBER" คำสั่ง RANDOMIZE; นี้จะช่วยให้เกมเริ่มต้นไม่เหมือนเดิม ซึ่งจะทำให้ผู้เล่นจับทิศทางไม่ได้ หลังจากนั้นเราก็โหลดเอาภาพตัวเลขมาจากไฟล์ "SHBD01" และเซตค่าต่าง ๆ โปรแกรมนี้เมนลูปจะอยู่ที่ MAINLOOP ดังนี้

```
MAINLOOP:
    CHECKENDPATTERN;
    IF SOLVED = 1 THEN GOTO EOJ1;
    IF SCORE < 1 THEN GOTO EOJ;
    UPDATETIMECOUNT;
    IF (T1 < 1) THEN GOTO EOJ;
    IF KEYPRESSED THEN GOTO
    READKBD;
    GOTO MAINLOOP;
```

คือมันจะหมุนวนอยู่ตรงนี้จนกว่าจะมีการกดแป้นคีย์อันใดอันหนึ่ง เมื่อกดเข้าไปแล้ว (KEYPRESSED จะเป็น TRUE) มันก็จะกระโดดไปที่ READKBD: ซึ่งจะมีการเช็คว่าเป็นตัว "E" หรือไม่ หรือเป็นตัวอักษรที่อยู่บนลูกบอล ถ้าเป็นตัวอักษรบนลูกบอลก็จะกระโดดต่อไปที่ READKBD2: ตรงนั้นเราจะทำการย้ายลูกบอลให้ ย้ายแล้วก็จะวนกลับไปที่ MAINLOOP: อีก ผมคิดว่าส่วนที่เหลือท่านคงจะอ่านเองได้

สรุปแล้วโปรแกรมนี้ยาวประมาณ 6 หน้ากระดาษ ผมใช้บางส่วนมาจากโปรแกรมอื่นที่มีอยู่แล้ว ท่านผู้อ่านคงจะสังเกตได้ เช่น โพรซีเดอร์ด้านการเรียกเวลามาจากดอส, การโหลดภาพตัวเลข, การสร้างภาพตัวเลข... เหล่านี้ใช้ชื่อเดิม ๆ ซึ่งเหมือนกับหลาย ๆ โปรแกรมที่ผมเคยนำมาลงในวารสารนี้ วิธีนี้ทำให้ผมทำโปรแกรมได้เร็วขึ้น แต่อย่างไรก็ตาม โปรแกรมขนาด 6 หน้ากระดาษนี้ถ้าเรามีความชำนาญในภาษานั้น ๆ เราจะเขียนเสร็จได้ (รวมทั้งดีบักและทดสอบ) ใน 2-3 วัน เกมนี้ผมลองเล่นดูแล้วก็เพลิดเพลินดีพอสมควร การกำหนดชุดตัวอักษรที่ไม่ซ้ำเดิมนี้ช่วยให้เกมสนุกยิ่งขึ้น ท่านผู้อ่านที่มีพีซีชนิดเอทีคอมแพท มีจอสีและมีเมมโมรีไม่น้อยกว่า 512 กิโลไบต์ ก็จะเล่นเกมนี้ได้ โปรแกรมนี้ชื่อ SHBS01.PAS มันมีออปเจ็กต์ชื่อ SHBS01.COM มันต้องใช้ไฟล์ชื่อ SHBD01 เข้าช่วย และจะต้องมีไลบรารี GRAPH.P และ GRAPH.BIN มาช่วยในตอนคอมพายล์ โปรแกรมนี้พอจะใช้เป็นตัวอย่างแสดงการใช้คำสั่งกราฟิกในเทอร์โบปาสคาลเวอร์ชัน 3.0 ได้ ท่านที่อ่านดูแล้วติดปัญหาหรือสงสัยตรงไหนก็โทรไปปรึกษาได้ (โทร. 311-4767 หรือ 332-3223)

รูปที่ 2 ตัวอย่างโปรแกรม

```
PROGRAM SHBS01;
{$I GRAPH.P }
LABEL STARTINGPOINT,LOOP1,MAINLOOP,READKBD,READKBD2,EOJ1,EOJ;
TYPE  ST06=STRING[6];
      ST176=ARRAY[0..176] OF BYTE;
      PICREC=RECORD PCS:ST06; PIC:ST176;END;
VAR CH:CHAR;
VAR CHESSPICFILE:FILE OF PICREC;
VAR CHESSPICREC:PICREC;
VAR ST,ST0,ST1,BNS:ST176;
VAR NPAL:INTEGER;
VAR DIGI:ARRAY[0..9] OF ST176;
VAR L1,L2,K,SHFT,SOLVED:INTEGER;
VAR PIECE:ST06;
VAR SCORE,DEDUCTION:INTEGER;
VAR SCR:ARRAY[1..5] OF INTEGER;
VAR TAB:ARRAY[1..9] OF INTEGER;
VAR HOUR,MIN,SEC,FRAG,T0,T1,T2,SVT,SPF:INTEGER;

PROCEDURE GETTIME;
TYPE REGPACK=RECORD
               AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS:INTEGER;
               END;
VAR REGS:REGPACK;
BEGIN
    WITH REGS DO
      BEGIN
      AX:=$2C00;
      MSDOS(REGS);
      HOUR:=HI(CX);
      MIN:=LO(CX);
      SEC:=HI(DX);
      FRAG:=LO(DX);
      END;
END;

PROCEDURE UPDATETIMECOUNT;
BEGIN
  GETTIME;T2:=60*MIN+SEC;
  IF T2<T0 THEN T2:=T2+3600;
  T1:=1800-T2+T0;
  IF T1<>SVT THEN
     BEGIN
     GOTOXY(20,3);WRITE('TIME=',T1:4);
     SVT:=T1;
     END;
END;

PROCEDURE LOADPICTURE;
VAR K:INTEGER;
BEGIN
  ASSIGN(CHESSPICFILE,'SHBD01');
  RESET(CHESSPICFILE);
  CLEARSCREEN;
  SEEK(CHESSPICFILE,1);
  READ(CHESSPICFILE,CHESSPICREC);
  ST0:=CHESSPICREC.PIC;
  FOR K:=0 TO 9 DO
    BEGIN
    SEEK(CHESSPICFILE,K+2);
    READ(CHESSPICFILE,CHESSPICREC);
    DIGI[K]:=CHESSPICREC.PIC;
    END;
  SEEK(CHESSPICFILE,43);
  READ(CHESSPICFILE,CHESSPICREC);
  BNS:=CHESSPICREC.PIC;
  CLOSE(CHESSPICFILE);
END;

PROCEDURE SPEEDCHECK;
VAR X,T1,T2,K1,K2:INTEGER;
BEGIN
  X:=0;
  GETTIME;
  T1:=100*SEC+FRAG;
  FOR K1:=1 TO 2000 DO
    FOR K2:=1 TO 30 DO
      BEGIN X:=X;END;
  GETTIME;
  T2:=100*SEC+FRAG;
  IF T2<T1 THEN T2:=T2+6000;
  SPF:=25000 DIV (T2-T1);
END;

PROCEDURE TIMEOUT(L:INTEGER);
VAR K1,K2:INTEGER;
BEGIN
  FOR K1:=1 TO L DO
    FOR K2:=1 TO SPF DO
      BEGIN SPF:=SPF;END;
END;

PROCEDURE DRAWBOX(X1,Y1,X2,Y2,C:INTEGER);
BEGIN
  DRAW(X1,Y1,X1,Y2,C);
  DRAW(X2,Y1,X2,Y2,C);
  DRAW(X1,Y1,X2,Y1,C);
  DRAW(X1,Y2,X2,Y2,C);
END;

PROCEDURE BEEP;
BEGIN
  SOUND(650);TIMEOUT(10);NOSOUND;TIMEOUT(10);
END;

PROCEDURE PING;
BEGIN
  SOUND(2300);TIMEOUT(10);NOSOUND;TIMEOUT(10);
END;

PROCEDURE PRINTSCORE(SC:INTEGER);
VAR K,L:INTEGER;
BEGIN
  SCR[1]:=SC DIV 10000;SC:=SC-10000*SCR[1];
  SCR[2]:=SC DIV 1000;SC:=SC-1000*SCR[2];
  SCR[3]:=SC DIV 100;SC:=SC-100*SCR[3];
  SCR[4]:=SC DIV 10;SC:=SC-10*SCR[4];
  SCR[5]:=SC;
  FOR K:=1 TO 5 DO
```

รูปที่ 2 (ต่อ)

```
    BEGIN
    L:=SCR[K];
    PUTPIC(DIGI[L],24*K-23,23);
    END;
END;

PROCEDURE NEWSCREEN;
VAR K1,K2:INTEGER;
BEGIN
  CLEARSCREEN;
  PRINTSCORE(SCORE);
  FOR K2:=24 TO 199 DO
   BEGIN
   PLOT(0,K2,1);
   PLOT(319,K2,1);
   TIMEOUT(1);
   END;
  FOR K1:=0 TO 319 DO
   BEGIN
   PLOT(K1,24,1);
   PLOT(K1,199,1);
   TIMEOUT(1);
   END;
  GOTOXY(20,2);WRITE('BALL MOVING GAME');
  GOTOXY(20,3);WRITE('TIME=',T1:4);
  GOTOXY(30,23);WRITE('E=END');
END;

PROCEDURE GENCHAR;
VAR X1,Y1,X2,Y2,X3,Y3,K1,K2:INTEGER;
BEGIN
  SHFT:=5+RANDOM(14);
  FOR K1:=1 TO 9 DO
    BEGIN
    X1:=24*(K1-1)+47;Y1:=89;
    X2:=X1+24;Y2:=Y1+24;
    X3:=X1;Y3:=Y2;
    DRAWBOX(X1,Y1,X2,Y2,1);
    CIRCLE(X1+12,Y1+12,12,1);
    FILLSHAPE(X1+13,Y1+13,3,1);
    GOTOXY(3*K1+5,3+10);
    CH:=CHR(64+K1+SHFT);
    TAB[K1]:=64+K1+SHFT;
    IF (K1=9) THEN
       BEGIN CH:=' ';TAB[K1]:=0;END;
    WRITE(CH);
    DRAWBOX(X1+8,Y1+6,X1+16,Y1+14,0);
    UPDATETIMECOUNT;
    END;
END;

PROCEDURE MOVEBALL(L1,L2:INTEGER);
VAR X1,Y1,X2,Y2,X3,Y3:INTEGER;
VAR X4,Y4,X5,Y5,X6,Y6:INTEGER;
BEGIN
    X1:=24*(L1-1)+47+1;Y1:=89+1;
    X2:=X1+24-1;Y2:=Y1+24-1;
    X3:=X1;Y3:=Y2;
    X4:=24*(L2-1)+47+1;Y4:=89+1;
    X5:=X4+24-1;Y5:=Y4+24-1;
    X6:=X4;Y6:=Y5;
    GETPIC(ST1,X1,Y1,X2,Y2);
    PUTPIC(ST1,X6,Y6);
    PUTPIC(ST0,X3,Y3-1);
    TAB[L2]:=TAB[L1];
    TAB[L1]:=0;
END;

PROCEDURE SCRAMBLECHAR;
VAR K1,K2,R1,R2:INTEGER;
VAR CH:CHAR;
BEGIN
  FOR K1:=1 TO 100 DO
    BEGIN
    REPEAT
      R1:=RANDOM(9)+1;
      UNTIL TAB[R1]>0;
    FOR K2:=1 TO 9 DO
      IF TAB[K2]=0 THEN R2:=K2;
    MOVEBALL(R1,R2);
    TIMEOUT(20);
    UPDATETIMECOUNT;
    PING;
    END;
END;

PROCEDURE GETL1L2(CH:CHAR;VAR L1,L2:INTEGER);
VAR K2:INTEGER;
BEGIN
  FOR K2:=1 TO 9 DO
    BEGIN
    IF CH=CHR(TAB[K2]) THEN L1:=K2;
    IF TAB[K2]=0 THEN L2:=K2;
    END;
END;

PROCEDURE CHECKENDPATTERN;
VAR K1,K2:INTEGER;
BEGIN
  SOLVED:=1;
  FOR K1:=1 TO 9 DO
    IF (K1<>9) THEN
       IF TAB[K1]<>64+K1+SHFT THEN SOLVED:=0;
END;

BEGIN
  SPEEDCHECK;
  GRAPHCOLORMODE;
  RANDOMIZE;
  PALETTE(1);
  LOADPICTURE;

STARTINGPOINT:
  SCORE:=2029;T1:=0;
  GETTIME;T0:=60*MIN+SEC;
  GETTIME;T2:=60*MIN+SEC;
  IF T2<T0 THEN T2:=T2+3600;
  T1:=1800;SVT:=T1;

LOOP1:
  NPAL:=RANDOM(4);
  PALETTE(NPAL);
  NEWSCREEN;
  GENCHAR;TIMEOUT(100);
  SCRAMBLECHAR;TIMEOUT(100);
  DEDUCTION:=15;

MAINLOOP:
  CHECKENDPATTERN;
  IF SOLVED=1 THEN GOTO EOJ1;
  IF SCORE<1 THEN GOTO EOJ;
  UPDATETIMECOUNT;
  IF (T1<1) THEN GOTO EOJ;
  IF KEYPRESSED THEN GOTO READKBD;
  GOTO MAINLOOP;

READKBD:
  READ(KBD,CH);
  IF CH='E' THEN GOTO EOJ;
  IF (CH>=CHR(64+SHFT+1)) AND
     (CH<=CHR(64+SHFT+9-1)) THEN GOTO READKBD2;
  SCORE:=SCORE-30;
  PRINTSCORE(SCORE);
  BEEP;
  GOTO MAINLOOP;

READKBD2:
  GETL1L2(CH,L1,L2);
  MOVEBALL(L1,L2);
  SCORE:=SCORE-DEDUCTION;
  DEDUCTION:=DEDUCTION+DEDUCTION DIV 2;
  PRINTSCORE(SCORE);
  BEEP;
  GOTO MAINLOOP;

EOJ1:
  FOR K:=1 TO 5 DO
    BEGIN
    GOTOXY(30,3);WRITE('YOU DID IT');
    SOUND(500);TIMEOUT(50);
    SOUND(600);TIMEOUT(50);
    GOTOXY(30,3);WRITE('          ');
    NOSOUND;
    END;
  SCORE:=SCORE+1250;
  PRINTSCORE(SCORE);
  GOTO LOOP1;

EOJ:
  FOR K:=1 TO 5 DO
    BEGIN
    GOTOXY(30,3);WRITE('END OF JOB');
    SOUND(1500);TIMEOUT(50);
    SOUND(1600);TIMEOUT(50);
    GOTOXY(30,3);WRITE('          ');
    NOSOUND;
    END;
  GOTOXY(30,3);WRITE('END OF JOB');
  END.
```